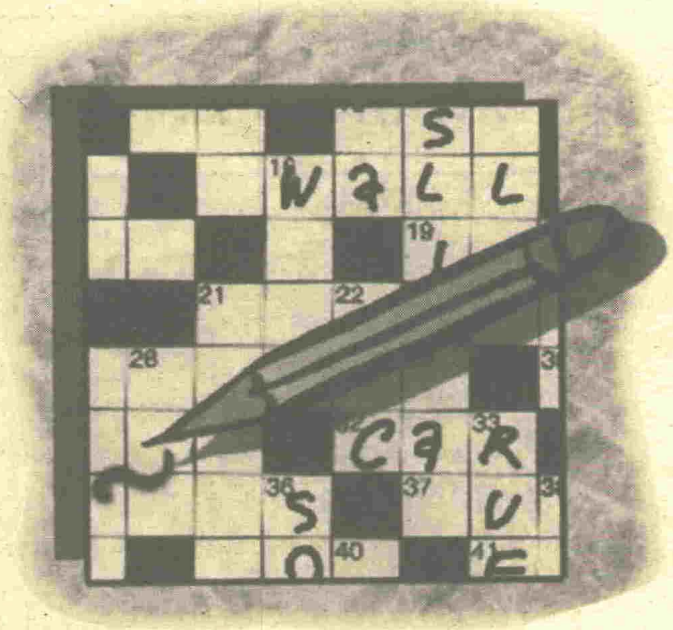
ขจร โรจนเมธินทร์

Visual BASIC
: ทำความรู้จักพรินเตอร์ออปเจ็กต์



ตัวอย่างการจัดการเรื่องการพิมพ์ ซึ่งแสดงให้เห็นการใช้งาน ของพรินเตอร์ออปเจ็กต์

พรินเตอร์เป็นออปเจ็กต์ชนิดหนึ่งในหลาย ๆ ตัวที่วิชวลเบสิกเตรียมมาให้ใช้ การใช้งานออปเจ็กต์ชนิดนี้มีหลักการที่ควรทราบอยู่หลายอย่าง สำหรับพรินเตอร์ออปเจ็กต์ของวิชวลเบสิก 4.0 ก็ยังคล้ายกับในรุ่นก่อน มีสิ่งที่ต่างกันเล็กน้อย ดังนั้นโค้ดเดิมที่เขียนด้วยวิชวลเบสิก 3.0 ก็ยังนำมาใช้งานได้

ปกติแล้วการทำงานของเครื่องพิมพ์จะถูกควบคุมโดยวินโดวส์ ดังนั้นพรินเตอร์ออปเจ็กต์จึงทำหน้าที่ส่งคำสั่งต่าง ๆ ไปให้รูทีนของวินโดวส์ หลังจากนั้นวินโดวส์จะส่งเอาต์พุตต่าง ๆ ไปยังเครื่องพิมพ์เอง หลักการของพรินเตอร์ออปเจ็กต์จะต่างไปจากการเขียนโปรแกรมเก่า ๆ หากผู้อ่านเคยเขียนโปรแกรมเพื่อควบคุมเครื่องพิมพ์คงจำได้ว่า เราต้องจัดการเอาต์พุตแบบซีเควนเชียล พูดง่าย ๆ ก็คือ หากต้องการพิมพ์เอกสารหนึ่งหน้า ก็จะต้องสั่งให้เครื่องพิมพ์พิมพ์ไปทีละบรรทัด ไม่สามารถย้อนกลับไปยังบรรทัดก่อนหน้าได้ สำหรับเอกสารที่มีแต่ตัวอักษรนั้น วิธีการพิมพ์ทีละบรรทัดดูจะใช้งานได้ดี แต่ถ้าเอกสารนั้นซับซ้อนมากขึ้น เช่น แต่ละหน้าของวารสารไมโครฯที่คุณกำลังอ่านนี้ จะเห็นว่า เฉพาะตัวอักษรยังแบ่งออกเป็นสองคอลัมน์ นอกจากนี้ยังมีรูปภาพประกอบในแต่ละหน้าอีก วิธีการพิมพ์แบบเดิมจะจัดการเอาต์พุตแบบนี้ได้ยาก แต่เมื่อใช้วิชวลเบสิกซึ่งมีวิธีจัดการการพิมพ์ต่างไปจากเดิม คือ ในการพิมพ์เอกสารหนึ่งหน้า เราสามารถจัดวางสิ่งต่าง ๆ ที่จะพิมพ์ไว้บนหน้ากระดาษสมมติได้ โดยจะวางที่ตรงไหนก่อนก็ได้ เช่น เราอาจวางตำแหน่งรูปภาพที่มุมบนซ้ายของหน้าก่อน แล้วสั่งให้วางตำแหน่งข้อความเฉพาะคอลัมน์ทางซ้าย จากนั้นย้อนกลับไปวางข้อความคอลัมน์ทางขวาได้ เมื่อวางตำแหน่งส่วนประกอบต่าง ๆ บนหน้าเรียบร้อยแล้ว จึงจะสั่งให้พิมพ์ออกทางเครื่องพิมพ์จริง

พรอพเพอร์ตี้ที่สำคัญของพรินเตอร์ออปเจ็กต์

ColorMode (VB 4.0)

ColorMode เป็นพรอพเพอร์ตี้ส์ที่มีในวิชวลเบสิก 4.0 ใช้กำหนดชนิดการพิมพ์แบบขาวดำหรือสี ไมโครซอฟต์เตรียมพรอพเพอร์ตี้ส์นี้มาเพื่อรองรับการพิมพ์แบบสีที่กำลังได้รับความนิยมเพิ่มขึ้น รูปแบบคำสั่งดังตารางที่ 1

Printer.ColorMode [= type%]		
ค่า	ชื่อค่าคงที่	ความหมาย
1	vbPRCMMonochrome	การพิมพ์แบบขาวดำ
2	vbPRCMColor	การพิมพ์แบบสี

ตารางที่ 1 แสดงค่าและความหมายของพรอพเพอร์ตี้ส์ ColorMode

CurrentX, CurrentY

CurrentX และ CurrentY เป็นพรอพเพอร์ตี้ส์ที่ใช้กำหนดตำแหน่งที่เราต้องการวางสิ่งที่จะพิมพ์บนหน้ากระดาษในแนวนอน (แกน X) และแนวตั้ง (แกน Y) ค่าที่กำหนดให้พรอพเพอร์ตี้ส์นี้จะต้องอยู่ในขอบเขตที่เป็นไปได้ หากผู้ใช้กำหนดค่าไปอยู่นอกกระดาษ สิ่งที่จะพิมพ์ก็จะพิมพ์ออกมาไม่ได้ นอกจากนี้เรายังสามารถอ่านค่าตำแหน่งปัจจุบันของพรินเตอร์ออปเจ็กต์ได้ ค่าที่อ่านได้จะขึ้นอยู่กับโหมดโคออร์ดิเนตที่เรากำหนดในพรอพเพอร์ตี้ส์ ScaleMode รูปแบบมีดังนี้

```
Printer.CurrentX [= xPosition!]
Printer.CurrentY [= yPosition!]
```

DeviceName, DriverName (VB 4.0)

พรอพเพอร์ตี้ส์ DeviceName ใช้สำหรับอ่านชื่อเครื่องพิมพ์ ซึ่งผู้ใช้ติดตั้งอยู่กับวินโดวส์ เช่น Canon Bubble-Jet BJ-330, Epson LQ-870 ESC/P2 เป็นต้น ส่วนพรอพเพอร์ตี้ส์ DriverName ใช้อ่านชื่อไดรเวอร์ที่เครื่องพิมพ์ซึ่งติดตั้งอยู่ใช้งาน เช่น EPSON24, EPSON9 เป็นต้น

Duplex (VB 4.0)

พรอพเพอร์ตี้ส์นี้จะใช้กำหนดให้เครื่องพิมพ์พิมพ์เอกสารบนกระดาษทั้งสองหน้า แต่จะใช้ได้เมื่อเครื่องพิมพ์ที่ใช้อยู่สนับสนุนการพิมพ์สองหน้า ผู้ใช้สามารถกำหนดวิธีให้เครื่องพิมพ์พลิกกระดาษไปพิมพ์หน้าที่สองในแนวนอนหรือแนวดิ่งก็ได้ รูปแบบการใช้ดังตารางที่ 2

FontCount, FontName, FontSize

พรอพเพอร์ตี้ส์ที่เกี่ยวข้องกับฟอนต์ของเครื่องพิมพ์ยังมีอีกหลายตัว ซึ่งคล้ายกับของคอนโทรลตัวอื่น ๆ เช่น FontBold, FontItalic, FontStrikethru, FontUnderline เป็นต้น แต่ที่กล่าวถึง 3 ตัวนี้ เพื่อจะเน้นการใช้งาน โดยพรอพเพอร์ตี้ส์ FontCount จะใช้อ่านจำนวนฟอนต์ทั้งหมดที่จะใช้พิมพ์ทางเครื่องพิมพ์ได้ ดังนั้นจึงเป็นพรอพเพอร์ตี้ส์ที่อ่านค่าได้อย่างเดียว FontName ใช้กำหนดชนิดฟอนต์ที่จะพิมพ์ออกทางเครื่องพิมพ์ โดยผู้ใช้ต้องเลือกชื่อฟอนต์ให้ถูกต้องเพื่อให้ได้แบบที่ต้องการ ส่วน FontSize ใช้กำหนดขนาดฟอนต์ที่จะใช้พิมพ์ว่าจะใหญ่โตขนาดไหน รูปแบบการใช้พรอพเพอร์ตี้ส์มีดังนี้

```
Printer.FontName [= fontname$]
Printer.FontSize [= fontsize%]
Height, Width
```

Height และ Width ของพรินเตอร์ออปเจ็กต์จะคล้ายกับฟอร์ม ต่างกันที่พรอพเพอร์ตี้ส์ Height และ Width ของพรินเตอร์ออปเจ็กต์จะเป็นค่าคงที่ โดยค่าจะขึ้นกับไดรเวอร์ของเครื่องพิมพ์แต่ละชนิดที่ใช้อยู่ อาจจะเรียกว่าค่าแสดงขนาดกระดาษทางฟิสิกคัลของเครื่องพิมพ์ เราไม่สามารถกำหนดค่าได้โดยตรง แต่ยังสามารถอ่านค่าขึ้นมาใช้ได้ ค่าของพรอพเพอร์ตี้ส์จะเปลี่ยนแปลงตามที่กำหนดในไดอะล็อก Printer Setup

Printer.Duplex [= value%]		
ค่า	ชื่อค่าคงที่	ความหมาย
1	vbPRDPSimplex	พิมพ์แบบหน้าเดียว
2	vbPRDPHorizontal	พิมพ์แบบสองหน้าโดยพลิกกระดาษในแนวนอน
3	vbPRDPVertical	พิมพ์แบบสองหน้าโดยพลิกกระดาษในแนวดิ่ง

ตารางที่ 2 แสดงค่าและความหมายของพรอพเพอร์ตี้ส์ Duplex

Printer.Orientation [= setting%]		
ค่า	ชื่อค่าคงที่	ความหมาย
1	vbPRORPortrait	พิมพ์ตามแนวนอนของกระดาษ
2	vbPRORLandscape	พิมพ์ตามแนวยาวของกระดาษ

ตารางที่ 3 แสดงค่าและความหมายของพรอพเพอร์ตี้ส์ Orientation

Orientation (VB 4.0)

พรอพเพอร์ตี้ส์นี้มีเพิ่มมาใหม่ในวิชวลเบสิก 4.0 ทำให้เราสามารถควบคุมการพิมพ์ได้มีประสิทธิภาพมากขึ้น ผู้ใช้สามารถกำหนดให้การพิมพ์ทำตามแนวนอนของกระดาษ (Portrait) หรือพิมพ์ตามแนวดิ่ง (Landscape) ก็ได้ โดยกำหนดค่าตามตารางที่ 3

Page

Page เป็นพรอพเพอร์ตี้ส์ที่วิชวลเบสิกเตรียมไว้ใช้นับหน้า โดยจะคืนค่าหน้าปัจจุบันให้ พรอพเพอร์ตี้ส์นี้จะอ่านค่าได้ขณะโปรแกรมทำงานเท่านั้น ไม่สามารถกำหนดขณะออกแบบโปรแกรมได้ ค่าของ Page จะเพิ่มขึ้นทีละหนึ่งทุกครั้งที่มีการเรียกเมธอด NewPage หรือกรณีที่เอกสารที่กำลังพิมพ์ด้วยเมธอด Print ไม่สามารถพิมพ์ได้พอดีในหนึ่งหน้าค่า Page ก็จะถูกเพิ่มขึ้นด้วย

PaperSize (VB 4.0)

PaperSize เป็นพรอพเพอร์ตี้ส์ใหม่อีกตัวที่เพิ่มขึ้นมาให้ใช้กำหนดหรืออ่านขนาดของกระดาษที่ใช้อยู่ เพื่อจะนำมาคำนวณหาบริเวณที่จะพิมพ์เอกสารได้ถูกต้องมากขึ้น ขนาดกระดาษที่ผู้ใช้กำหนดได้มีถึง 41 แบบ เช่น Letter 8 1/2 x11 in, Ledger 17 x 11 in, Statement 5 1/2 x 8 1/2 in, Envelope#9 3 7/8 x 8 7/8 in เป็นต้น

PrintQuality (VB 4.0)

พรอพเพอร์ตี้ส์นี้ใช้กำหนดความละเอียดของสิ่งที่จะพิมพ์ว่าจะให้ละเอียดระดับไหน ตั้งแต่แบบ draft ถึงความละเอียดสูงสุด ตามค่าในตารางที่ 4 แต่หากผู้ใช้ไม่พอใจกับความละเอียดที่วิชวลเบสิกกำหนดมา ก็อาจกำหนดเป็นเลขบวกที่แทนความละเอียดในหน่วย dpi ก็ได้ เช่น 300 dpi

ค่า	ชื่อค่าคงที่	ความหมาย
-1	vbPRPQDraft	พิมพ์แบบ draft
-2	vbPRPQLow	พิมพ์ความละเอียดต่ำสุด
-3	vbPRPQMedium	พิมพ์ความละเอียดระดับกลาง
-4	vbPRPQHigh	พิมพ์ความละเอียดสูงสุด

ตารางที่ 4 แสดงค่าและความหมายของพรอพเพอร์ตี้ส์ PrintQuality

ค่า	หน่วยวัด	ความหมาย
0	User-defined	ผู้ใช้กำหนดค่าเอง
1	Twip	1 นิ้ว มีทั้งหมด 1440 twips
2	Point	1 นิ้วมี 72 points
3	Pixel	จุดขนาดเล็กสุดเท่าที่เครื่องพิมพ์จะทำได้
4	Character	หนึ่งคาแรคเตอร์จะสูง 1/6 นิ้ว กว้าง 1/12 นิ้ว
5	Inch	หน่วยเป็นนิ้ว
6	Millimeter	หน่วยเป็นมิลลิเมตร
7	Centimeter	หน่วยเป็นเซนติเมตร

ตารางที่ 5 แสดงชนิดของหน่วยวัดที่สามารถกำหนดได้ใน ScaleMode

Printer.Circle [Step](Xpos!, Ypos!), Radius![, Color&][, Start!][, End!][, Aspect!]

อาร์กิวเมนต์	ความหมาย
Step	กำหนดให้จุดศูนย์กลางขยับไปตามค่าของ Xpos! และ Ypos! เมื่อเทียบกับจุดปัจจุบัน
Xpos!, Ypos!	กำหนดจุดศูนย์กลางของวงกลม
Radius!	กำหนดขนาดรัศมี
Start!	กำหนดมุมเริ่มต้นเป็นเรเดียน กรณีวาดเส้นโค้ง
End!	กำหนดมุมสิ้นสุดเป็นเรเดียน กรณีวาดเส้นโค้ง
Aspect!	กำหนดสัดส่วนของวงในกรณีต้องการให้วาดวงรี

ตัวอย่างที่ 1

ScaleHeight, ScaleWidth

เป็นพรอพเพอร์ตี้ส์ที่จะใช้กำหนดหรืออ่านค่าขนาดที่ขอบภายในของออปเจ็กต์เมื่อจะใช้เมธอดทางกราฟิก ถ้าจะพูดให้เข้าใจง่าย ๆ ในกรณีของพรินเตอร์ออปเจ็กต์ก็คือ พรอพเพอร์ตี้ส์นี้จะแทนขนาดซึ่งเป็นบริเวณที่เราสามารถพิมพ์เอกสารได้ ดังนั้นในวิชวลเบสิก 3.0 เราจึงสามารถใช้พรอพเพอร์ตี้ส์นี้มาคำนวณได้ เพราะไม่มีพรอพเพอร์ตี้ส์ PaperSize ให้อ่านค่า พรอพเพอร์ตี้ส์ ScaleHeight จะเป็นขนาดในแนวนอน ส่วน ScaleWidth เป็นขนาดในแนวดิ่งเหมือนของฟอร์ม

ScaleLeft, ScaleTop

พรอพเพอร์ตี้ส์สองอันนี้จะคล้ายกับ ScaleHeight และ ScaleWidth แต่ความหมายของมันต่างกันเล็กน้อย คือแทนที่ค่าของพรอพเพอร์ตี้ส์จะแทนค่าของมุมล่างขวาของออปเจ็กต์ พรอพเพอร์ตี้ส์สองอันนี้จะแทนค่าที่มุมบนซ้ายของออปเจ็กต์ ซึ่งโดยมากแล้วค่าพรอพเพอร์ตี้ส์ทั้งสองจะถูกเซตเป็นศูนย์เสมอ แต่หากผู้ใช้ต้องการแก้ไขก็สามารถกำหนดใหม่ได้

ScaleMode

เป็นพรอพเพอร์ตี้ส์ซึ่งใช้กำหนดหน่วยวัดที่จะใช้กับระบบโคออร์ดิเนตของเครื่องพิมพ์ หน่วยวัดนี้จะเหมือนกับของฟอร์ม ผู้ใช้เลือกกำหนดค่าได้ตามตารางที่ 5

Zoom (VB 4.0)

ใช้กำหนดขนาดของเอาต์พุตที่จะพิมพ์เป็นเปอร์เซ็นต์เทียบกับขนาดจริง ผู้ใช้สามารถย่อหรือขยายผลงานที่จะพิมพ์ได้ตามชอบใจ การกำหนดค่า Zoom ทำได้ขณะโปรแกรมกำลังทำงาน

เมธอดที่สำคัญของพรินเตอร์ออปเจ็กต์

Circle

เมธอดนี้ใช้สั่งให้วาดวงกลม วงรี เส้นโค้งบนหน้ากระดาษพิมพ์ รูปแบบคำสั่งเป็นดังตัวอย่างที่ 1

EndDoc

พรอพเพอร์ตี้ส์นี้ใช้ส่งสัญญาณให้วินโดวส์ทราบว่าเราจัดวางตำแหน่งสิ่งที่ต้องการพิมพ์บนหน้ากระดาษสมมติเรียบร้อยแล้ว วินโดวส์ก็จะจัดการพิมพ์เอาต์พุตออกทางเครื่องพิมพ์ทันที พรอพเพอร์ตี้ส์มีความสำคัญตรงที่ผู้ใช้ต้องเรียกทุกครั้งเมื่อจัดตำแหน่งของสิ่งพิมพ์เรียบร้อยแล้ว มิฉะนั้นจะไม่มีอะไรพิมพ์ออกไปทางเครื่องพิมพ์เลย

KillDoc (VB 4.0)

เมธอด KillDoc นี้มีเพิ่มมาในวิชวลเบสิก 4.0 เพื่อให้ผู้ใช้สามารถยกเลิกงานที่เครื่องพิมพ์กำลังทำอยู่ทันที ดังนั้นกรณีที่เราไม่ต้องการให้พิมพ์เอกสารนั้นต่อก็สั่งได้ทันทีแทนที่ต้องรอให้งานนั้นพิมพ์เสร็จก่อน

Line

เราสามารถใช้เมธอด Line นี้ลากเส้นตรงหรือวาดสี่เหลี่ยมออกทางเครื่องพิมพ์บนหน้ากระดาษได้ โดยใช้คำสั่งดังตารางที่ 2

NewPage

เมธอดนี้ใช้ส่งสัญญาณให้วินโดวส์ทราบว่าหน้าปัจจุบันที่กำลังวางตำแหน่งส่วนประกอบต่าง ๆ อยู่เสร็จแล้ว และต้องการขึ้นหน้าใหม่เพื่อจะวางส่วนประกอบต่าง ๆ ในหน้าใหม่นี้ต่อไป ลักษณะการใช้ NewPage จะคล้ายกับ form feed ที่เราต้องสั่งเมื่อต้องการให้ขึ้นหน้าใหม่

PaintPicture (VB 4.0)

เมธอด PaintPicture ลดความยุ่งยากในการพิมพ์ภาพกราฟิกออกทางเครื่องพิมพ์ได้มาก เพราะสามารถใช้เมธอดนี้วาดกราฟิกในไฟล์ฟอร์แมต .BMP, .WMF, .EMF, .ICO, .DIB การใช้คำสั่งนี้มีรูปแบบดังตัวอย่างที่ 3

Printer.Line [Step](StartX!, StartY!) - [Step](EndX!, EndY!)[, Color&][, B|, BF]

อาร์กิวเมนต์	ความหมาย
Step	จุดเริ่มต้นของเส้นตรงจะขยับไปตามค่า StartX!, StartY! เทียบกับจุดปัจจุบัน
StartX!, StartY!	จุดเริ่มต้นของเส้นตรง
Step	จุดสิ้นสุดของเส้นตรงจะขยับไปตามค่า EndX!, EndY! เมื่อเทียบกับจุดปัจจุบัน
EndX!, EndY!	จุดสิ้นสุดของเส้นตรง
Color&	กำหนดสีเส้นตรงที่จะพิมพ์
B	วาดสี่เหลี่ยมโดยใช้ StartX!, StartY! เป็นจุดมุมซ้ายบน และ EndX!, EndY! เป็นมุมขวาล่าง
BF	วาดสี่เหลี่ยมแต่ระบายด้วยสีที่กำหนดใน Color&

ตัวอย่างที่ 2

Printer.PaintPicture
(picture)(, x1, y1)[, width1][, height1][, x2, y2][, width2][, height2][, opcode]

อาร์กิวเมนต์	ความหมาย
Picture	เป็นภาพกราฟิกที่จะพิมพ์ซึ่งกำหนดไว้ในพรอพเพอร์ตี้ส์ Picture ของ Form หรือ PictureBox
x1, y1	จุดที่เริ่มต้นวาดภาพ
width1	ความกว้างของกราฟิกที่จะวาด ในกรณีที่ขนาดภาพไม่เท่ากับความกว้างที่กำหนด กราฟิกจะถูกยืดหรือย่อให้พอดี
height1	ความสูงของกราฟิกที่จะวาด กรณีที่ขนาดภาพไม่เท่ากับความกว้างที่กำหนด กราฟิกจะถูกยืดหรือย่อให้พอดี
x2, y2	จุดของบริเวณที่จะเริ่มคลิบปิ่งภาพใน picture ถ้าไม่กำหนดถือว่าเป็น 0
width2	ความกว้างใน picture ที่จะเริ่มคลิบปิ่งภาพ ถ้าไม่กำหนดจะใช้ขนาดความกว้างของ picture
height2	ความสูงใน picture ที่จะเริ่มคลิบปิ่งภาพ ถ้าไม่กำหนดจะใช้ขนาดความสูงของ picture
opcode	โอเปอเรชันที่ใช้ทำบิตแม็พกับ picture เช่น Not, Xor

ตัวอย่างที่ 3

Printer.Print [{Spc(n)|Tab(n)}][expressionlist][{;|,}]

อาร์กิวเมนต์	ความหมาย
Spc(n)	กำหนดให้พิมพ์ช่องว่างทั้งหมด n ช่อง จากตำแหน่งที่พิมพ์ครั้งสุดท้ายก่อน
Tab(n)	กำหนดให้พิมพ์แท็บไป m ครั้งก่อน
expressionlist	สิ่งที่จะพิมพ์
;	กำหนดให้พิมพ์ข้อมูลต่อจากตำหน่งก่อนหน้าทันที
,	กำหนดให้พิมพ์ข้อมูลในตำแหน่งช่วงถัดไป ปกติแต่ละช่วงจะเท่ากับ 14 คอลัมน์

ตัวอย่างที่ 4

Print

เมธอด Print นี้จะเรียกว่าเป็นเมธอดสำคัญที่สุดก็ได้ เพราะใช้สำหรับพิมพ์เท็กซ์ไปที่พรินเตอร์ออปเจ็กต์โดยเฉพาะ การใช้เมธอด Print นี้โดยมากมักจะใช้พรอพเพอร์ตี้ส์ CurrentX และ CurrentY ประกอบด้วยเพื่อจัดวางตำแหน่งของเท็กซ์ที่ต้องการดังทีกล่าวมาแล้ว รูปแบบเป็นดังตัวอย่างที่ 4

Printer.Pset [Step](Xpos!, Ypos!)[, Color&]

อาร์กิวเมนต์	ความหมาย
Step	กำหนดให้จุดที่จะวาดขยับไปที่ Xpos!, Ypos! เทียบกับจุดที่อยู่ปัจจุบัน
Xpos!, Ypos!	จุดที่จะวาด
Color&	สีของจุดที่วาด

ตัวอย่างที่ 5

Printer.Scale (x1, y1) - (x2, y2)

อาร์กิวเมนต์	ความหมาย
x1, y1	จุดมุมบนซ้าย
x2, y2	จุดมุมล่างขวา

ตัวอย่างที่ 6

PrintForm

เป็นเมธอดที่ใช้ก๊อบปี้ส่วนประกอบต่าง ๆ ที่อยู่ในฟอร์มที่ทำงานอยู่ส่งต่อให้กับรูทีนที่ทำหน้าที่พิมพ์ของวินโดวส์ ส่วนต่าง ๆ ที่ปรากฏบนจอภาพจะถูกส่งออกพิมพ์ รายละเอียดของภาพที่ได้จะใกล้เคียงกับความจริง รูปแบบคำสั่งมีดังนี้

```
[form.]PrintForm
```

Pset

เมธอดนี้จะใช้สั่งให้วาดจุดลงบนหน้าที่กำลังพิมพ์อยู่ รูปแบบคำสั่งดังตัวอย่างที่ 5

Scale (VB 4.0)

ใช้กำหนดระบบโคออร์ดิเนตให้กับพรินเตอร์ออปเจ็กต์ เมธอด Scale จะช่วยให้ผู้ใช้กำหนดสเกลได้ตามต้องการ (ดูตัวอย่างที่ 6)

TextHeight

ใช้หาความสูงของเท็กซ์ที่จะส่งให้เครื่องพิมพ์ เมธอดนี้มีความสำคัญเพราะจะต้องเรียกใช้เพื่อคำนวณพื้นที่ซึ่งจะพิมพ์บนหน้ากระดาษที่ถูกต้อง โดยเฉพาะจะใช้ตรวจสอบว่าเท็กซ์บรรทัดที่กำลังจะพิมพ์เลยขอบล่างของกระดาษหรือไม่

```
Printer.TextHeight(text$)
```

TextWidth

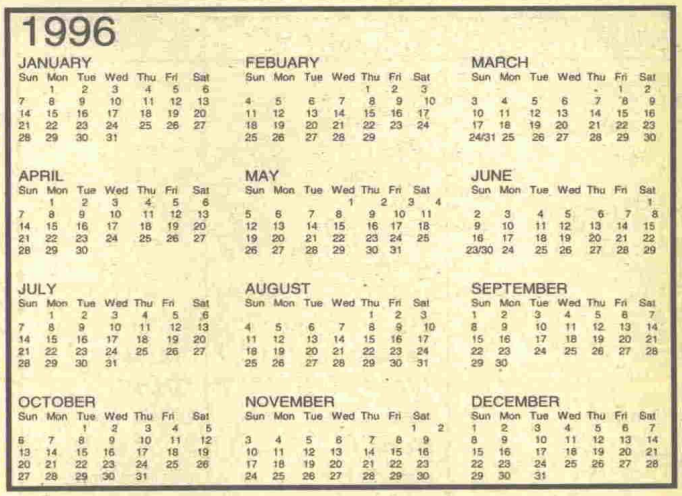
ใช้หาความกว้างของเท็กซ์ที่จะพิมพ์เมธอด TextWidth นี้มักจะต้องนำมาช่วยในการคำนวณว่าการพิมพ์ครั้งต่อไปจะเกินขอบขวาของหน้ากระดาษหรือยัง

```
Printer.TextWidth(text$)
```

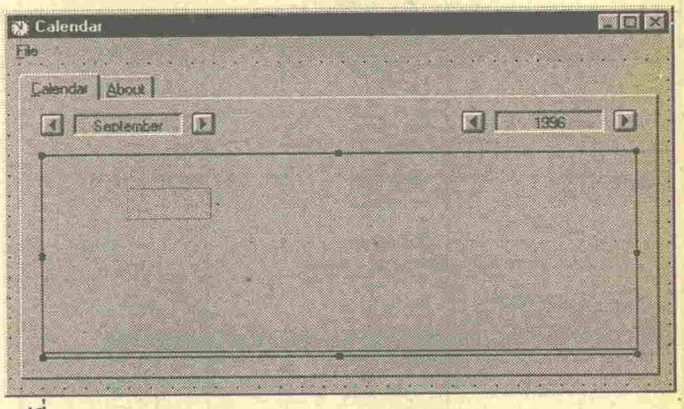
ตัวอย่างพรินเตอร์โปรเจ็กต์

พรินเตอร์โปรเจ็กต์ที่ใช้เป็นตัวอย่างแสดงการใช้งานพรินเตอร์ออปเจ็กต์ โปรแกรมที่ใช้เป็นตัวอย่างนี้นำมาจากโปรแกรม VB Calendar ที่เคยลงในไมโครฉบับที่ 113 ซึ่งใช้แสดงปฏิทินแบบ 100 ปี แต่ว่าผู้เขียนนำโปรแกรมมาปรับปรุงต่อโดยเพิ่มความสามารถให้โปรแกรมพิมพ์ปฏิทินปี คศ. ที่โปรแกรมแสดงอยู่ในแบบ 3 คอลัมน์ได้ (ดูรูปที่ 1)

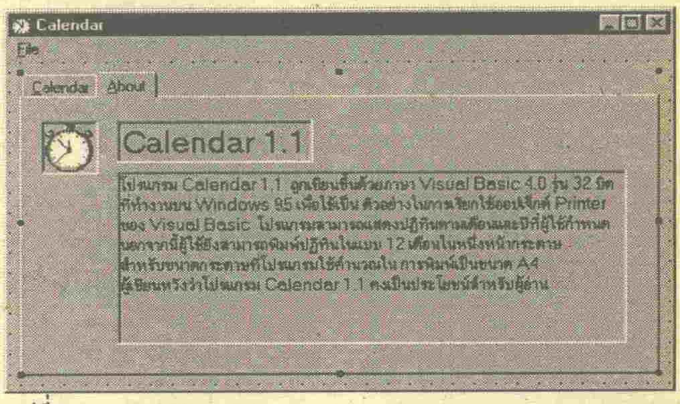
พรินเตอร์โปรเจ็กต์นี้ผู้เขียนออกแบบบนวิชวลเบสิก 4.0 รุ่น 32 บิต บนวินโดวส์ 95 ดังนั้นจึงนำอินเตอร์เฟสของวินโดวส์ 95 เช่น แท็บชีตมาใช้ จากรูปที่ 2 แสดงส่วนประกอบของคอนโทรลที่ใช้ในโปรแกรม ผู้เขียนเลือกคอนโทรลแท็บชีต SSTab1 มาวางเป็นพื้นของฟอร์มที่ออกแบบ โดยตั้งชื่อแท็บชีตที่หนึ่งเป็น Calendar และแท็บชีตที่สองชื่อ About (ดูรูปที่ 3) เราสามารถกำหนดจำนวนแท็บที่พรอพเพอร์ตี้ส์ Tabs สำหรับพรอพเพอร์ตี้ส์ TabsPerRow ใช้กำหนดจำนวนแท็บต่อหนึ่งแถว นอกจากนี้เราจะเลือกสไตล์ของแท็บชีตให้เป็นแบบวินโดวส์ 95 พรอพเพอร์ตี้ส์ Style ส่วนประกอบที่วางบนแท็บชีตอันแรกก็คือ กริดคอนโทรล grdMonth ที่ใช้สำหรับแสดงปฏิทิน โดยเราจะกำหนดสีแบคกราวนด์ให้เป็นสีเทาเพื่อให้กลืนไปกับสีของแท็บชีต และบังคับไม่ให้กริดมีกรอบ เส้นตารางในกริด และแถบสีในแสดงตำแหน่งเซล กำหนดจำนวนในกริดแถวเป็น 6 และคอลัมน์เป็น 7 แต่ในโปรแกรมนี้ผู้เขียนตัดเมนูส่วนที่ใช้กำหนดเดือนและปีที่ต้องการให้โปรแกรมแสดงปฏิทินมาใช้เป็นคอนโทรลเลเบลสองอันวางไว้บนแท็บชีต คือ lblYear และ lblMonth โดยกำหนดให้พรอพเพอร์ตี้ส์ Alignment ให้เป็น Center จากนั้นจึงใช้คอมมานด์บัททอนแบบสามมิติวางไว้ที่หน้าและท้ายเลเบลทั้งสองอัน แต่ในที่นี้ผู้เขียนต้องการรูปลูกศรมาแสดงบนบัททอนเพื่อให้ผู้ใช้คลิ๊กเพื่อเพิ่มและลดเดือนหรือปีที่แสดงในเลเบล สำหรับรูปลูกศรนี้ผู้เขียนใช้ Paintbrush วาดเป็นไฟล์ BMP ก่อนแล้วจึงโหลดลงบนคอมมานด์บัททอนอีกที ส่วนแท็บชีตอันที่สองใช้อิมเมจคอนโทรลแสดงรูปไอคอน เลเบลคอนโทรล 2 อัน คือ Label1 และ Label2 แสดงข้อความอธิบายโปรแกรม สำหรับพรอพเพอร์ตี้ส์ Caption ของคอนโทรล Label1 นั้นเป็นข้อความยาวมาก วิชวลเบสิกเก็บข้อความนี้รวมอยู่กับไอคอนและบิตแม็พต่าง ๆ ไว้ในไฟล์ calendar.frx สำหรับเมนูของโปรแกรม มีเมนูย่อยที่สำคัญคือ Print เพื่อใช้พิมพ์ปฏิทิน (ดูตารางที่ 6)



รูปที่ 1



รูปที่ 2



รูปที่ 3

โค้ดที่ต้องเขียนในโปรแกรม

เราจะเริ่มกันที่โค้ดส่วนซึ่งกำหนดตัวแปรก่อน (รูปที่ 4) MonthName และ DayName เป็นอะเรย์ใช้เก็บสตริงชื่อเดือนและวันในสัปดาห์ ส่วน grdMnth และ grdYear ใช้เก็บเดือนและปีของปฏิทินที่แสดงอยู่ในกริด xpos เป็นอะเรย์ที่ใช้เก็บตำแหน่งชื่อวันทั้งเจ็ดที่พิมพ์เป็นปฏิทินแต่ละเดือน ส่วนตัวแปร FntName และ FntSize ใช้เก็บชื่อและขนาดฟอนต์ที่เลือกใช้สำหรับพิมพ์ปฏิทิน

```
Option Explicit
Dim MonthName(12) As String
Dim DayName(7) As String
Dim grdMnth As Integer
Dim grdYear As Integer
Dim xpos(7) As Integer
Dim FntName As String
Dim FntSize As Integer
```

รูปที่ 4

ออปเจ็กต์	พรอพเพอร์ตี้ส์	ค่าเริ่มต้น
VB.Form	Name	Form1
	BorderStyle	1 'Fixed Single
	Caption	"Calendar"
	MaxButton	0 'False
TabDlg.SSTab	Name	SSTab1
	Caption	"&Calendar"
	TabsPerRow	2
	Tabs	2
	Style	1
	TabCaption(0)	"&Calendar"
	TabCaption(1)	"&About"
VB.Label	Name	Label2
	Alignment	2 'Center
	BorderStyle	1 'Fixed Single
	Caption	"Calendar 1.1"
VB.Label	Name	Label1
	BorderStyle	1 'Fixed Single
VB.Image	Name	Image1
	BorderStyle	1 'Fixed Single
	Stretch	-1 'True
Threed.SSCommand	Name	b3dYrInc
Threed.SSCommand	Name	b3dYrDec
VB.Label	Name	lblYear
	Alignment	2 'Center
	BorderStyle	1 'Fixed Single
	Caption	"1996"
Threed.SSCommand	Name	b3dMnDec
Threed.SSCommand	Name	b3dMnInc
VB.Label	Name	lblMonth
	Alignment	2 'Center
	BorderStyle	1 'Fixed Single
	Caption	"September"
MSGrid.Grid	Name	grdMonth
	BackColor	12632256 'Gray
	FontName	"Brush Script MT"
	FontSize	14.25
	BorderStyle	0
	Rows	6
	Cols	7
	FixedRows	0
	FixedCols	0
	GridLines	0 'False
	HighLight	0 'False
VB.Menu	Name	mnuFile
	Caption	"&File"
VB.Menu	Name	mnuPrint
	Caption	"&Print"
VB.Menu	Name	mnuSep
	Caption	"-"
VB.Menu	Name	mnuExit
	Caption	"E&xit"

ตารางที่ 6 แสดงค่าเริ่มต้นของคอนโทรลแต่ละอันที่เป็นส่วนประกอบของฟอร์ม

```
Sub InitDayName()
        DayName(1) = "Sun"
        DayName(2) = "Mon"
        DayName(3) = "Tue"
        DayName(4) = "Wed"
        DayName(5) = "Thu"
        DayName(6) = "Fri"
        DayName(7) = "Sat"
End Sub

Sub InitMonthName()
        MonthName(1)  = "January"
        MonthName(2)  = "Febuary"
        MonthName(3)  = "March"
        MonthName(4)  = "April"
        MonthName(5)  = "May"
        MonthName(6)  = "June"
        MonthName(7)  = "July"
        MonthName(8)  = "August"
        MonthName(9)  = "September"
        MonthName(10) = "October"
        MonthName(11) = "November"
        MonthName(12) = "December"
End Sub
```

รูปที่ 5

โค้ดในรูปที่ 5 แสดงซับรูทีน InitDayName และ InitMonthName ซึ่งใช้กำหนดค่าเริ่มต้นเป็นสตริงชื่อเดือนให้ตัวแปรอะเรย์ DayName และ MonthName เพื่อความสะดวกในการเรียกใช้ภายหลัง

ฟังก์ชันในรูปที่ 6 ทำหน้าที่เหมือนเดิมคือ DayOfMonth จะคืนค่าจำนวนวันในเดือนของปีที่ต้องการทราบให้ ส่วนฟังก์ชัน Find1stDay จะใช้ค้นว่าเดือนและปีที่กำหนดมาในอาร์กิวเมนต์ mm, yy นั้นวันที่ 1 ของเดือนตรงกับวันอะไร ฟังก์ชันจะคืนค่าเป็นตัวเลขคือ 1 (วันอาทิตย์) ถึง 7 (วันเสาร์) สาเหตุที่ต้องหาว่าวันที่ 1 ของเดือนที่ต้องการเป็นวันอะไรก็เพื่อจะสร้างปฏิทินขึ้นได้ถูกต้อง

โค้ดในรูปที่ 7 เป็นซับรูทีนที่เกี่ยวข้องในการแสดงปฏิทินบนกริด ผู้เขียนเปลี่ยนวิธีกำหนดค่าแต่ละเซลในกริดใหม่ ขอเริ่มที่ซับรูทีน FillDayName ที่จะนำชื่อวันที่ที่เก็บใน DayName มาใส่ในเซลแถวที่ 1 ทั้ง 7 คอลัมน์ทีละคอลัมน์ โดยเริ่มกำหนดให้ grdMonth.Row เป็น 0 จากนั้นใช้ลูป For...Next กำหนดค่า grdMonth.Col เพื่อเลื่อนไปทีละคอลัมน์เพื่อนำสตริงของอะเรย์มากำหนดค่าพรอพเพอร์ตี้ส์ Text ของกริด สาเหตุที่ต้องกำหนดพรอพเพอร์ตี้ส์ FixedAlignment เพราะต้องการใช้ชื่อวันแสดงอยู่กึ่งกลางของเซล เนื่องจากก่อนจบซับรูทีนนี้เราจะกำหนดให้แถวแรกของกริดเป็นแถวที่อยู่คงที่ ซับรูทีน SetGridAlign มีหน้าที่กำหนดให้ข้อมูลที่แสดงอยู่ในกริดทุกคอลัมน์แสดงอยู่กึ่งกลางเซล โดยกำหนดค่าพรอพเพอร์ตี้ส์ ColAlignment ของทุกคอลัมน์ให้เป็น 2

ซับรูทีนสำคัญที่จะกล่าวถึงต่อไปนี้ คือ FillCalendar ที่ทำหน้าที่สร้างปฏิทินที่ผู้ใช้ต้องการขึ้นมาแสดงในกริด grdMonth จากรูปที่ 8 เป็นตัวอย่างปฏิทินของเดือนมีนาคม ปี 2539 ซึ่งในที่นี้ผู้เขียนจำกัดให้ปฏิทินที่แสดงจะมีได้แค่ 5 แถว และ 7 คอลัมน์ โดยด้านซ้ายสุดเป็นวันอาทิตย์ และขวาสุดเป็นวันเสาร์ การทำงานของซับรูทีนจะเริ่มต้นด้วยการเรียกฟังก์ชัน DayOfMonth เพื่อหาว่าเดือนที่ผู้ใช้จะสร้างปฏิทินมีกี่วัน ในที่นี้จะมีค่า 31 และเก็บจำนวนวันไว้ในตัวแปร numOfDay เราจะใช้ลูป

```
Function DayOfMonth(mm, yy As Integer) As Integer
  Select Case mm
    Case 1, 3, 5, 7, 8, 10, 12
      DayOfMonth = 31
    Case 4, 6, 9, 11
      DayOfMonth = 30
    Case 2
      DayOfMonth = 28
      If (yy Mod 4) = 0 Then DayOfMonth = 29
  End Select
End Function

Function Find1stDay(mm, yy As Integer) As Integer
  Dim i As Integer
  Dim j As Integer

  i = (yy + (yy - 1) \ 4) Mod 7
  Select Case mm
    Case 2, 3, 11
      i = (i + 3) Mod 7
    Case 4, 7
      i = (i + 6) Mod 7
    Case 5
      i = (i + 1) Mod 7
    Case 6
      i = (i + 4) Mod 7
    Case 8
      i = (i + 2) Mod 7
    Case 9, 12
      i = (i + 5) Mod 7
    Case 10
      i = (i + 7) Mod 7
  End Select
  j = yy Mod 4
  If (j = 0) And (mm > 2) Then
    i = (i + 1) Mod 7
  End If
  If i = 0 Then i = 7
  Find1stDay = i
End Function
```

รูปที่ 6

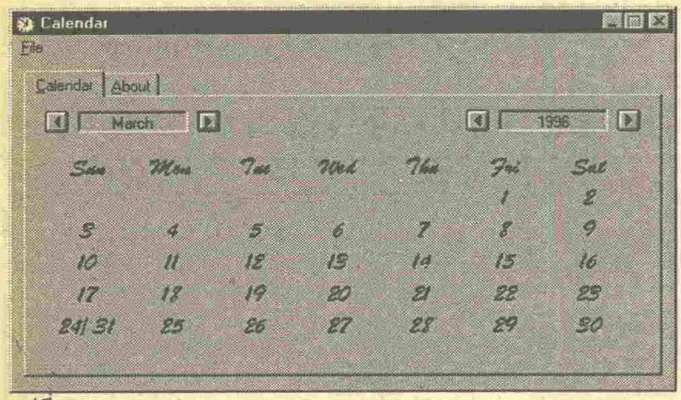
Do...While อันแรกเพื่อนับว่าใส่ตัวเลขวันที่ของแต่ละเดือนครบหรือยัง โดยใส่ตัวเลขไปทีละแถว ตัวแปร grdRow จะใช้สำหรับนับแถว ตัวเลขจะถูกใส่ในเซลทีละคอลัมน์ของวันในแต่ละแถว ด้วยลูป Do...While อันที่สอง ตัวแปร k จะถูกใช้เป็นอินเด็กซ์สำหรับเซลของกริดแต่ละคอลัมน์ จากนั้นจึงเริ่มสร้างปฏิทินสัปดาห์แรก ซึ่งโปรแกรมได้ค้นหาว่าวันที่ 1 ของเดือนมีนาคมต้องกับวันอะไรด้วยการเรียกฟังก์ชัน Find1stDay (กรณีที่ firstWeek เป็นจริง) ในที่นี้คือวันศุกร์ ดังนั้นในแถวแรกก็จะต้องใส่เลข 1 ในคอลัมน์ที่ 5 (ช่องวันศุกร์) และเลข 2 ในช่องวันเสาร์ ในกรณีที่ไม่ใช้สัปดาห์แรก ตัวเลขจะเริ่มใส่ที่คอลัมน์ที่ 0 เสมอ ดังนั้นกรณีที่เดือนนั้นมี 31 วันและวันแรกเป็นวันศุกร์ วันที่ 31 จะใส่ในเซลของกริดไม่ได้เพราะเรากำหนดให้กริดแสดงปฏิทิน 5 แถวเท่านั้น ดังนั้นกรณีนี้จึงใส่เลขวันที่ 31 ไว้ในแถวที่ 5 แทนเป็น 24/31

ซับรูทีน ClearCalendar ใช้ลบปฏิทินที่แสดงอยู่ โดยเราจะเซตค่าของแต่ละเซลในกริดโดยกำหนดค่าให้ grdMonth.Text เป็น “”

ซับรูทีนสำคัญสำหรับพิมพ์ปฏิทินที่แสดงในรูปที่ 9 คือ PrintMonth ซับรูทีนนี้ทำหน้าที่พิมพ์ปฏิทินทีละเดือน อาร์กิวเมนต์ที่ต้องส่งให้ซับรูทีนคือ x ใช้กำหนดตำแหน่งที่จะพิมพ์ปฏิทินในแนวนอนบนกระดาษ mm เป็นเดือนที่จะพิมพ์ day1st เป็นตำแหน่งวันในสัปดาห์ของวันที่ 1 ของเดือนนั้น ซึ่งเราได้ค่าจากฟังก์ชัน Find1stDay ซับรูทีนนี้

```
Sub ClearCalendar()
  Dim i As Integer
  Dim j As Integer
  For i = 1 To grdMonth.Rows - 1
    For j = 0 To grdMonth.Cols - 1
      grdMonth.col = j
      grdMonth.row = i
      grdMonth.Text = ""
    Next
  Next
End Sub

Sub FillCalendar()
  Dim numOfDay As Integer
  Dim firstWeek As Integer
  Dim grdRow As Integer
  Dim grdCols As Integer
  Dim i As Integer
  Dim j As Integer
  Dim k As Integer

  firstWeek = True
  grdRow = 1
  grdCols = grdMonth.Cols - 1
  i = 1
  j = 1
  numOfDay = DayOfMonth(grdMnth, grdYear)
  Do While (i <= numOfDay)
    If firstWeek Then
      j = Find1stDay(grdMnth, grdYear) - 1
      firstWeek = False
    Else
      j = 0
    End If
    k = j
    If grdRow > grdMonth.Rows - 1 Then
      grdRow = grdRow - 1
    End If
    grdMonth.row = grdRow
    Do While (k <= grdCols)
      grdMonth.col = k
      If grdMonth.Text = "" Then
        grdMonth.Text = i
      Else
        grdMonth.Text = grdMonth.Text + "/" + Str(i)
      End If
      i = i + 1
      If i > numOfDay Then Exit Do
      k = k + 1
    Loop
    grdRow = grdRow + 1
  Loop
End Sub

Sub FillDayName()
  Dim i As Integer

  grdMonth.row = 0
  For i = 0 To grdMonth.Cols - 1
    grdMonth.FixedAlignment(i) = 2 'Center
    grdMonth.col = i
    grdMonth.Text = DayName(i + 1)
  Next
  grdMonth.FixedRows = 1
End Sub

Sub SetGridAlign()
  Dim i As Integer
  For i = 0 To grdMonth.Cols - 1
    grdMonth.ColAlignment(i) = 2
  Next
End Sub
```

รูปที่ 7



รูปที่ 8

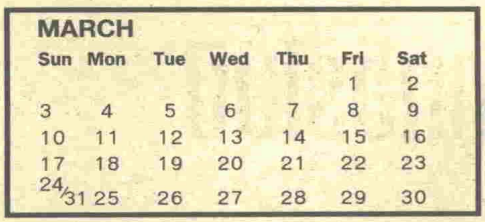
เป็นหัวใจสำคัญในการพิมพ์ปฏิทิน การทำงานเริ่มจากจะต้องหาจำนวนวันในเดือนด้วยฟังก์ชัน DayOfMonth จากนั้นจึงกำหนดชนิดและขนาดฟอนต์ที่จะใช้กับพรินเตอร์ออปเจ็กต์ กำหนดตำแหน่งที่จะพิมพ์ในแนวนอนผ่านพรอพเพอร์ตี้ส์ CurrentX เราจะเริ่มพิมพ์ชื่อเดือนด้วยตัวอักษรตัวใหญ่ โดยใช้ฟังก์ชัน Ucase$ (ดูรูปที่ 10) และพิมพ์ชื่อของวันในบรรทัดถัดไป โดยนำสตริงจากอะเรย์ DayName มาใช้ สำหรับตำแหน่งของชื่อวันซึ่งเราจะพิมพ์จะจัดให้ห่างกันคงที่โดยใช้ความกว้างของเท็กซ์ "Wed" เป็นหลัก ตำแหน่งที่พิมพ์ชื่อวันนี้จะถูกเก็บในอะเรย์ xpos เพื่อจะนำไปใช้ตอนพิมพ์ตัวเลขวันที่อีกครั้ง เมื่อพิมพ์แถวชื่อวัน (Sun...Sat) เสร็จเราก็จะเริ่มพิมพ์ตัวเลขวันที่ ตำแหน่งของการพิมพ์ตัวเลขจะนำมาจาก xpos แล้วกำหนดให้กับ Printer.CurrentX สาเหตุที่ต้องกำหนดตำแหน่งด้วยค่าจาก xpos ก็เพื่อให้ตัวเลขอยู่ตรงกับชื่อวัน เพราะถ้าไม่กำหนดเช่นนี้ปฏิทินที่ได้จะเบี้ยวไม่ต้องกัน เนื่องจากขนาดฟอนต์แต่ละตัวไม่ได้เท่ากันคงที่ ปฏิทินที่พิมพ์จะแยกเป็นสองกรณี คือ กรณีที่ไม่ใช่สัปดาห์ที่ 5 ของเดือนก็จะพิมพ์เลขวันด้วยเมธอด Printer.Print j; เพื่อให้การพิมพ์อยู่ในบรรทัดเดียวกัน จนกว่าจะพิมพ์ไปครบสัปดาห์คือ 1 เป็น 7 เราจะขึ้นบรรทัดใหม่และเพิ่มค่า row ทีละหนึ่ง กรณีที่พิมพ์ปฏิทินถึงสัปดาห์ที่ 5 ของเดือน โปรแกรมต้องจัดการพิมพ์พิเศษต่างไปจากสัปดาห์อื่น โดยเราจะพิจารณาที่คอลัมน์ 1 และ 2 หรือที่วันอาทิตย์และวันจันทร์ เพราะถ้าวันอาทิตย์ของสัปดาห์ที่ 5 เป็นวันที่ 23 แล้วโปรแกรมต้องพิมพ์เป็น 23/30 แทน หรือถ้าเป็นวันที่ 24 ก็ต้องพิมพ์เป็น 24/31 แทน โดยก่อนพิมพ์ก็ต้องปรับตำแหน่งใหม่ให้ขยับไปข้างหน้าเท่ากับความกว้างของ "0"

โค้ดในรูปที่ 11 แสดงซับรูทีน Print12 ซึ่งมีหน้าที่พิมพ์ปฏิทินทั้ง 12 เดือนออกทางพรินเตอร์ออปเจ็กต์ รูปแบบของปฏิทินที่จะพิมพ์จะเป็นตามรูปที่ 1 คือ พิมพ์เป็น 4 แถว แถวละ 3 เดือน การพิมพ์ปฏิทินแต่ละเดือนจะเป็นหน้าที่ของซับรูทีน PrintMonth สิ่งที่จะต้องทำในซับรูทีน Print12 ก็คือจัดตำแหน่งที่จะพิมพ์แต่ละเดือนให้ถูกต้องเพื่อส่งเป็นอาร์กิวเมนต์ให้ PrintMonth ใช้ได้ถูกต้อง จากรูปที่ 11 เราต้องเรียกเมธอด Print ก่อนหนึ่งครั้ง เนื่องจากเป็นบักของพรินเตอร์ออปเจ็กต์ในวิชวลเบสิก 4.0 มิฉะนั้นการพิมพ์จะไม่ได้ฟอนต์ตามที่กำหนด จากนั้น

```
Sub PrintMonth(x As Integer, mm As Integer,
day1st As Integer)
  Dim i As Integer
  Dim j As Integer
  Dim numOfDay As Integer
  Dim col As Integer
  Dim row As Integer

  numOfDay = DayOfMonth(mm, grdYear)
  Printer.FontName = FntName
  Printer.FontSize = FntSize + 3
  Printer.CurrentX = x
  Printer.Print UCase$(MonthName(mm))
  Printer.FontSize = FntSize
  Printer.CurrentX = x
  For i = 1 To 7
    xpos(i) = Printer.CurrentX
    Printer.Print DayName(i);
    Printer.CurrentX = Printer.CurrentX + Printer
   .TextWidth("Wed ") - Printer.TextWidth(DayName(i))
  Next
  Printer.Print
  i = 1
  j = 1
  col = 1
  row = 1
  Printer.CurrentX = x
  Do While j <= numOfDay
    Printer.CurrentX = xpos(col)
    If i >= day1st Then
      If row < 5 Then
        Printer.Print j;
      End If
      If row = 5 Then
        Select Case col
          Case 1, 2
            If (j = 23) Or (j = 24) Then
              If j + 7 <= numOfDay Then
                Printer.CurrentX = Printer.
                CurrentX - Printer.TextWidth("0")
                Printer.Print j; "/"; j + 7;
              Else
                Printer.Print j;
              End If
            Else
              Printer.Print j;
            End If
          Case Else
            Printer.Print j;
        End Select
      End If
      j = j + 1
    End If
    col = col + 1
    If col > 7 Then col = 1
    If i Mod 7 = 0 Then
      Printer.Print
      row = row + 1
    End If
    i = i + 1
  Loop
  If row = 5 Then Printer.Print
End Sub
```

รูปที่ 9

เราจะพิมพ์ปีคศ. โดยใช้ฟอนต์ขนาดใหญ่กว่าฟอนต์ที่พิมพ์ปฏิทินบวก 23 สำหรับการพิมพ์ปฏิทินแต่ละเดือนจะใช้ลูป For...Next 2 อัน โดยเราจะพิมพ์ทีละคอลัมน์โดยเริ่มจากคอลัมน์แรกก่อนคือเดือน January, April, July และ October จากนั้นจึงขยับขึ้นไปพิมพ์คอลัมน์ที่สองและสามตามลำดับโดยแต่ละคอลัมน์จะห่างกันเท่ากับความกว้างของเท็กซ์



MARCH

Sun	Mon	Tue	Wed	Thu	Fri	Sat
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24/31	25	26	27	28	29	30

รูปที่ 10

```
Sub Print12()
  Dim offsetX As Integer
  Dim offsetY As Integer
  Dim i As Integer
  Dim j As Integer
  Dim day1st As Integer
  Dim x As Integer

  Printer.Print                    'Bug in VB 4.0
  offsetX = (Printer.Width -Printer.ScaleWidth) \ 2
  offsetY = (Printer.Height-Printer.ScaleHeight)\2
  Printer.CurrentY = offsetY
  Printer.FontSize = FntSize + 23
  Printer.CurrentX = offsetX +Printer.TextWidth("Wed")
  Printer.Print grdYear
  offsetY = Printer.CurrentY
  For j = 0 To 2
    Printer.CurrentY = offsetY
    x = offsetX + xpos(7) +Printer.TextWidth("Wed ")
  For i = j + 1 To j + 10 Step 3
     day1st = Findlst Day(i, grdYear)
    PrintMonth x, i, day1st
    Printer.Print
    Printer.Print
   Next i
  Next j
   Printer.EndDoc
End Sub
```

รูปที่ 11

"Wed" เมื่อกำหนดพิมพ์สิ่งต่าง ๆ ผ่านพรินเตอร์ออปเจ็กต์แล้วก็ต้องเรียกเมธอด EndDoc เพื่อส่งสัญญาณให้วินโดวส์พิมพ์ปฏิทินออกทางเครื่องพิมพ์จริง ๆ

โค้ดในรูปที่ 12 เป็นส่วนที่รับผิดชอบเหตุการณ์ที่ผู้ใช้คลิ๊กบนบัททอนรูปลูกศรเพื่อกำหนดเดือนและปีที่ต้องการให้แสดงในปฏิทิน บัททอนมีทั้งหมด 2 กลุ่ม คือ กลุ่มลูกศรซ้ายใช้ลดค่า และกลุ่มลูกศรขวาใช้เพิ่มค่า สิ่งที่ต้องทำในแต่ละซับรูทีนจะคล้ายกันคือปรับค่าตัวแปร grdMnth หรือ grdYear ตามแต่กรณี จากนั้นแก้ไขพรอพเพอร์ตี้ส์ Caption ของเลเบล lblMonth หรือ lblYear จากนั้นก็ลบปฏิทินของเก่าด้วยซับรูทีน ClearCalendar และแสดงปฏิทินใหม่ด้วย FillCalendar

โค้ดในรูปที่ 13 ซับรูทีน Form_Load จะกำหนดค่าเริ่มต้นของตัวแปร grdMnth, grdYear รายละเอียดของฟอนต์ที่จะใช้พิมพ์ กำหนดค่าอะเรย์ MonthName และ DayName และจัดกำหนดตำแหน่งของเซล ซับรูทีน Form_Activate มีหน้าที่จัดให้ฟอร์มแสดงอยู่กลางจอภาพ กำหนดค่าเริ่มต้นของพรอพเพอร์ตี้ส์ Caption ของเลเบลทั้งสองที่อยู่บนฟอร์ม จากนั้นจึงแสดงปฏิทินของเดือนปัจจุบัน สำหรับโค้ดในรูปที่ 14 จะรับผิดชอบเหตุการณ์ในเมนู Print ที่เรียกซับรูทีน Print12 เพื่อพิมพ์ปฏิทิน 12 เดือน

การใช้เครื่องพิมพ์ในวิชวลเบสิกนั้นค่อนข้างสะดวก เพราะมีพรินเตอร์ออปเจ็กต์เตรียมมาให้เรียบร้อยแล้ว แทนที่จะต้องไปเรียกใช้

```
Private Sub b3dMnDec_Click()
    If grdMnth > 1 Then
        grdMnth = grdMnth - 1
        lblMonth.Caption = MonthName(grdMnth)
        ClearCalendar
        FillCalendar
    End If
End Sub

Private Sub b3dMnInc_Click()
    If grdMnth < 12 Then
        grdMnth = grdMnth + 1
        lblMonth.Caption = MonthName(grdMnth)
        ClearCalendar
        FillCalendar
    End If
End Sub

Private Sub b3dYrDec_Click()
    grdYear = grdYear - 1
    lblYear.Caption = grdYear
    ClearCalendar
    FillCalendar
End Sub

Private Sub b3dYrInc_Click()
    grdYear = grdYear + 1
    lblYear.Caption = grdYear
    ClearCalendar
    FillCalendar
End Sub
```

รูปที่ 12

```
Private Sub Form_Activate()
    Top = (Screen.Height - Height) \ 2
    Left = (Screen.Width - Width) \ 2
    lblMonth.Caption = MonthName(grdMnth)
    lblYear.Caption = grdYear
    FillDayName
    FillCalendar
End Sub

Private Sub Form_Load()
    grdMnth = Month(Date)
    grdYear = Year(Date)
    FntName = "Arial"
    FntSize = 7
    Printer.FontName = FntName
    Printer.FontSize = FntSize
    InitMonthName
    InitDayName
    SetGridAlign
End Sub
```

รูปที่ 13

```
Private Sub mnuExit_Click()
    End
End Sub

Private Sub mnuPrint_Click()
    Print12
End Sub
```

รูปที่ 14

ฟังก์ชันเอพีไอ Escape ที่ค่อนข้างยุ่งยาก จากตัวอย่างในบทความนี้แสดงให้เห็นว่าการจัดการเรื่องการพิมพ์เป็นเรื่องที่ละเอียดพอสมควร การที่จะได้ผลงานพิมพ์สวย ๆ ออกมาสักแผ่นหนึ่งก็ต้องใช้ความพยายามพอควร แต่ปัจจุบันมีแอพพลิเคชันอำนวยความสะดวกหลาย ๆ อย่าง เราจึงไม่ต้องลำบาก อย่างไรก็ตาม บทความนี้คงแสดงให้เห็นการใช้งานพรินเตอร์ออปเจ็กต์ได้บ้าง